

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 62-163149
(43)Date of publication of application : 18.07.1987

(51)Int.Cl. G06F 9/46

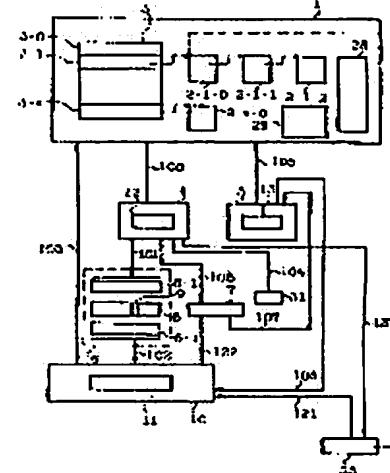
(21)Application number : 61-005391 (71)Applicant : HITACHI LTD
(22)Date of filing : 14.01.1986 (72)Inventor : TANAKA TOSHIHARU
UMENO HIDENORI

(54) DISPATCHING CONTROL SYSTEM FOR VIRTUAL COMPUTER

(57)Abstract:

PURPOSE: To reduce an overhead by selecting and starting a virtual processor that is run next based on the running priority designated by a VMCP.

CONSTITUTION: A PSW 8 is set under a waiting state by an OS on a running virtual processor. In other words, an instruction which sets a wait bit 9 at '1' is outputted to an instruction executing circuit 4 via a line 100. Thus the bit 9 of the PSW 8 is set at '1' via a line 101 in case said instruction can be executed directly by the circuit 4 and an instruction executing microprogram 12. Then a virtual processor state saving mechanism 10 is started by a line 102 to save the state of a virtual processor together with a virtual processor state saving microprogram 11. Thus it is possible to drive the virtual processor that satisfies the running conditions with an inactive processors excepting for the corresponding processor.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

⑫ 公開特許公報 (A)

昭62-163149

⑬ Int.CI.⁴

G 06 F 9/46

識別記号

350

厅内整理番号

8120-5B

⑭ 公開 昭和62年(1987)7月18日

審査請求 未請求 発明の数 5 (全 13 頁)

⑮ 発明の名称 仮想計算機のディスパッチ制御方式

⑯ 特願 昭61-5391

⑯ 出願 昭61(1986)1月14日

⑰ 発明者 田中俊治 川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

⑰ 発明者 梅野英典 川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

⑰ 出願人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

⑰ 代理人 弁理士 磯村雅俊

明細書

1. 発明の名称

仮想計算機のディスパッチ制御方式

2. 特許請求の範囲

(1) 1台の実計算機のもとで1以上のオペレーティングシステム(OS)の同時走行を仮想計算機制御プログラムにより制御する仮想計算機システムにおいて、上記仮想計算機を構成する仮想プロセッサごとに状態の追跡と、該仮想プロセッサがアクティブ状態か否かを示すアクティブフラグと、上記制御プログラムが指定する走行優先度とを格納する記憶領域、および各プロセッサがアクティブ状態か否かを示すプロセッサ状態表を有し、走行中のOSが該当プロセッサをウエイト状態にする命令を発行したとき、該命令を制御プログラムの介在なしにハードウェアおよびマイクロプログラムが直接実行した後、走行中の仮想プロセッサの状態を上記状態追跡領域に格納し、該仮想プロセッサのアクティブフラグに非アクティブ状態を

示し、さらに上記プロセッサ状態表に非アクティブ状態が指示されているプロセッサの中からプロセッサを選択して、プロセッサ状態表の該当フィールドにアクティブ状態を設定し、またウエイト状態でなく、かつアクティブ状態でない仮想プロセッサ群の中から走行優先度に従つて仮想プロセッサを選択するとともに、該仮想プロセッサのアクティブフラグにアクティブ状態を設定し、該仮想プロセッサの状態追跡領域の内容をプロセッサに設定することにより、仮想プロセッサを起動することを特徴とする仮想計算機のディスパッチ制御方式。

(2) 上記ハードウェアおよびマイクロプロセッサは、非アクティブ状態が指示されているプロセッサの中からプロセッサを選択した後、上記選択したプロセッサ上で走行させる仮想プロセッサが存在しない場合には、プロセッサ状態表の該プロセッサのフィールドに非アクティブ状態を指示することを特徴とする特許請求の範囲第1項記載の仮想計算機のディスパッチ制御方式。

(3) 1台の実計算機のもとで1以上のオペレーティングシステム(OS)の同時走行を仮想計算機制御プログラムにより制御する仮想計算機システムにおいて、上記仮想計算機を構成する仮想プロセッサごとに状態の追迹と、該仮想プロセッサがアクティブ状態か否かを示すアクティブフラグと、上記制御プログラムが指定する走行優先度とを格納する記憶領域、および各プロセッサがアクティブ状態か否かを示すプロセッサ状態表を有し、計算機のプロセッサにより第1の仮想プロセッサが走行中に、上記制御プログラムを介在せずにハードウェアおよびマイクロプログラムが直接処理可能な割込みが発生し、該割込みにより第2の仮想プロセッサのウェイト状態が解除された場合、第1の仮想プロセッサの状態を該当する状態追迹領域に格納し、また該仮想プロセッサのアクティブフラグに非アクティブ状態を設定するとともに、上記プロセッサおよびプロセッサ状態表に非アクティブ状態が指示されているプロセッサの中からプロセッサを選択して、プロセッサ状態表の該当

- 3 -

ムにおいて、上記仮想計算機を構成する仮想プロセッサごとに状態の追迹と、該仮想プロセッサがアクティブ状態か否かを示すアクティブフラグと、上記制御プログラムが指定する走行優先度と、上記制御プログラムが仮想プロセッサごとに指定したタイムスライス値とを格納する記憶領域、各プロセッサがアクティブ状態か否かを示すプロセッサ状態表、およびOSが仮想プロセッサにより走行開始後、上記タイムスライス値の時間が経過した時に、割込み信号を発生するタイマを有し、仮想プロセッサが走行開始後、タイムスライス値の時間が経過した時、ハードウェアおよびマイクロプログラムが走行中の該仮想プロセッサの状態を上記状態追迹領域に格納し、該プロセッサのアクティブフラグに非アクティブ状態を示し、さらに上記プロセッサ状態表に非アクティブ状態が指示されているプロセッサの中からプロセッサを選択するとともに、プロセッサ状態表の該当フィールドにアクティブ状態を設定し、またウェイト状態でなく、かつアクティブ状態でない仮想プロセッ

フィールドにアクティブ状態を設定し、またウェイト状態でなく、かつアクティブ状態でない仮想プロセッサ群の中から走行優先度に従って仮想プロセッサを選択して、該仮想プロセッサのアクティブフラグにアクティブ状態を設定し、上記選択したプロセッサの状態追迹領域の内容をプロセッサに設定することにより、仮想プロセッサを起動することを特徴とする仮想計算機のディスパッチ制御方式。

(4) 上記ハードウェアおよびマイクロプログラムは、非アクティブ状態が指示されているプロセッサを選択した後、選択されたプロセッサ上で走行させる仮想プロセッサが存在しない場合には、プロセッサ状態表の該プロセッサのフィールドに非アクティブ状態を指示することを特徴とする特許請求の範囲第3項記載の仮想計算機のディスパッチ制御方式。

(5) 1台の実計算機のもとで1以上のオペレーティングシステム(OS)の同時走行を仮想計算機制御プログラムにより制御する仮想計算機システ

- 4 -

サ群の中から走行優先度に従って仮想プロセッサを選択するとともに、該仮想プロセッサのアクティブフラグにアクティブ状態を設定し、該仮想プロセッサのタイムスライス値と該状態追迹領域の内容をそれぞれタイマとプロセッサに設定することにより、仮想プロセッサを起動することを特徴とする仮想計算機のディスパッチ制御方式。

(6) 上記ハードウェアおよびマイクロプログラムは、非アクティブ状態が指示されているプロセッサの中からプロセッサを選択した後、選択されたプロセッサ上で走行させる仮想プロセッサが存在しない場合には、プロセッサ状態表の該プロセッサのフィールドに非アクティブ状態を指示することを特徴とする特許請求の範囲第5項記載の仮想計算機のディスパッチ制御方式。

(7) 1台の実計算機のもとで1以上のオペレーティングシステム(OS)の同時走行を仮想計算機制御プログラムにより制御する仮想計算機システムにおいて、上記仮想計算機を構成する仮想プロセッサ群ごとに状態の追迹と、該仮想プロセッサが

アクティブ状態か否かを示すアクティブフラグと、上記制御プログラムが指定する走行優先度とを格納する記憶領域、各プロセッサがアクティブ状態か否かを示すプロセッサ状態表、および仮想プロセッサの選択・起動用命令を有し、走行中の仮想プロセッサの走行が中断され、上記制御プログラムまたはハードウェアおよびマイクロプログラムにより走行中の仮想プロセッサの状態を該当する仮想プロセッサの状態追跡領域に格納し、該仮想プロセッサのアクティブフラグに非アクティブ状態を設定した後、上記制御プログラムは上記選択・起動用命令を発行し、該命令は上記プロセッサおよびプロセッサ状態表に非アクティブ状態が指示されているプロセッサの中からプロセッサを選択するとともに、プロセッサ状態表の該当フィールドにアクティブ状態を設定し、またウエイト状態でなく、かつアクティブ状態でない仮想プロセッサ群の中から走行優先度に従つて仮想プロセッサを選択して、該仮想プロセッサのアクティブフラグにアクティブ状態を設定し、該状態追跡領域の

- 7 -

アクティブ状態か否かを示すプロセッサ状態表、1組以上の命令コードと条件コードの保持手段、マルチプロセッサの仮想計算機を構成する仮想プロセッサ群の状態判断手段、および状態の選択手段を有し、OSが走行中に、上記保持手段に示された命令を発行し、該命令に対する条件コードが上記保持手段に示された条件コードに一致するならば、ハードウェアおよびマイクロプログラムが走行中の仮想プロセッサの状態を上記状態追跡領域に格納して、該仮想プロセッサのアクティブフラグに非アクティブ状態を示し、さらに上記プロセッサ状態表に非アクティブ状態が指示されているプロセッサの中からプロセッサを選択するとともに、プロセッサ状態表の該当フィールドにアクティブ状態を設定し、また上記選択手段によりマルチプロセッサを構成する仮想プロセッサ群の状態の判定と選択を行い、選択した仮想プロセッサのアクティブフラグにアクティブ状態を設定し、選択した仮想プロセッサの状態追跡領域の内容を上記選択したプロセッサに設定することにより、

内容をプロセッサに設定することにより、仮想プロセッサを起動することを特徴とする仮想計算機のディスパッチ制御方式。

(8) 上記制御プログラムまたはハードウェアおよびマイクロプログラムは、非アクティブ状態が指示されているプロセッサの中からプロセッサを選択した後、選択されたプロセッサ上で走行させる仮想プロセッサが存在しない場合には、プロセッサ状態表の該プロセッサのフィールドに非アクティブ状態を指示することを特徴とする特許請求の範囲第7項記載の仮想計算機のディスパッチ制御方式。

(9) 1台の実計算機のもとで1以上のオペレーティングシステム(OS)の同時走行を仮想計算機制御プログラムにより制御する仮想計算機システムにおいて、上記仮想計算機を構成する仮想プロセッサごとに状態の追跡と該仮想プロセッサがアクティブ状態か否かを示すアクティブフラグと上記制御プログラムが指定する走行優先度とを格納する記憶領域、計算機を構成する各プロセッサが

- 8 -

走行仮想プロセッサを起動することを特徴とする仮想計算機のディスパッチ制御方式。

(10) 上記ハードウェアおよびマイクロプロセッサは、非アクティブ状態が指示されているプロセッサの中からプロセッサを選択した後、選択されたプロセッサ上で走行させる仮想プロセッサが存在しない場合には、プロセッサ状態表の該プロセッサのフィールドに非アクティブ状態を指示することを特徴とする特許請求の範囲第9項記載の仮想計算機のディスパッチ制御方式。

3. 発明の詳細な説明

【発明の利用分野】

本発明は、仮想計算機のディスパッチ制御方式に関し、特にマルチプロセッサの仮想計算機システムにおいて、各プロセッサを有効に利用するディスパッチ制御方式に関するものである。

【発明の背景】

仮想計算機システム (Virtual Machine System: 以下、VMSと略記する)は、1台の計算機のもとで、1つ以上のOS (オペレーティング

システム) の同時走行を可能とするシステムである。仮想計算機制御プログラム (Virtual Machine Control Program: 以下 V M C P と略記する) は、1台の実計算機上に各 O S ごとに、あたかも仮想計算機 (Virtual Machine: 以下、 V M と略記する) と呼ばれる計算機が存在するかのように制御するプログラムである。

V M S は、このように1つ以上の O S の同時走行を可能とするために、計算機のプロセッサを使用する V M 上の仮想プロセッサの切換処理を行う必要がある。従来は、 O S が発行した特権命令を V M C P がシミュレーションを行っていたが、高性能化のために、マイクロプログラムシミュレーションにより高速化したり、さらにはハードウェアにより直接実行する技術が開発されている。仮想計算機システムにおいて、次に走行する仮想プロセッサの選択と起動をソフトウェアにより行うと、そのオーバヘッドは大きい。

例えば、特開昭59-167756号公報には、仮想プロセッサが計算機上のあるプロセッサでア

イドル状態等になつた場合に、ハードウェアおよびマイクロプログラムにより仮想プロセッサを切換える方法が開示されている。この方法によれば、アイドル状態等になつた仮想プロセッサが動作していたプロセッサの有効利用が図られる。

しかし、上記の方法においては、 V M がマルチプロセッサ構成で、その中の第1の仮想プロセッサが同一 V M の他の仮想プロセッサのロットにより、 O S が第1の仮想プロセッサ上でループする場合のプロセッサ有効利用については、何も述べられていない。また、アイドル状態等になつた仮想プロセッサが動作していたプロセッサの有効利用については述べられているが、この時にアクティブ状態となつていない他のプロセッサの有効利用については、全く述べられていない。

計算機システムの中で処理される仕事の単位であるタスク(プロセスともいう)は、 C P U 以外の必要な全てのリソースが確保されると実行可能な状態になり、 C P U が割当てられるのを待つている。 C P U 以外のリソースを確保できないで待つ

- 11 -

ているタスクは、ウェイト状態にあるといわれる。実行可能な状態にあるタスクに C P U が割当てられると、アクティブ状態に入る。また、実行中のタスクが、優先度の高いタスクに C P U を奪われると、実行可能な状態に戻る。すなわち、タスクは、アクティブ状態、実行可能状態、およびウェイト状態の3つの状態のいずれかにあり、実行可能な状態にあるタスクに C P U を割当てるものをディスパッチャ(dispatcher)と呼ぶ。このディスパッチャは、 C P U を管理して、タスクの実行を制御するもので、 O S の中核をなす特殊な部分である。

【発明の目的】

本発明の目的は、このような従来の問題を改善し、 V M 上の仮想プロセッサがウェイト状態になつたり、同一 V M 上の他の仮想プロセッサのロットにより O S がループ状態となる場合には、その仮想プロセッサが走行しているプロセッサ、およびアクティブ状態にないプロセッサ上で、走行可能な仮想プロセッサを起動することができ、これ

によりプロセッサの有効利用を実現する仮想計算機のディスパッチャ制御方式を提供することにある。

【発明の概要】

上記目的を達成するため、本発明による仮想計算機のディスパッチャ制御方式は、 V M S の各 V M 上の仮想プロセッサごとに、プロセッサ状態の退避とアクティブフラグと V M C P が設定する走行優先度とタイムスライス値とを格納する領域を備え、また計算機を構成する各プロセッサがアクティブ状態か否かを示すプロセッサ状態表を備える。本発明による仮想プロセッサ切換処理を仮想プロセッサの切換え要因別に述べる。

(1) V M が走行中に、 V M 上の O S がプロセッサをウェイト状態にする命令を発行し、この命令を V M C P が介在することなく、ハードウェアおよびマイクロプログラムが直接実行した結果、仮想プロセッサがウェイト状態となつた場合には、仮想プロセッサ状態退避マイクロプログラムとともに、走行中の仮想プロセッサの状態を上記状態退避領域に格納し、またその仮想プロセッサのア

クティブフラグに、非アクティブ状態を設定する。次に、仮想プロセッサ状態追跡機構および仮想プロセッサ状態追跡マイクロプログラムは、主記憶装置上のディスパッチ命令群に制御を渡す。ディスパッチ命令群は、プロセッサ状態表に非アクティブ状態が指示されているプロセッサを選択し、プロセッサ状態表のそのプロセッサに対応するフィールドにアクティブ状態を指示する。また、ウエイト状態でなく、かつアクティブ状態でない仮想プロセッサ群の中で、最も走行優先度の高い仮想プロセッサを選択し、その仮想プロセッサのアクティブフラグにアクティブ状態を指示する。ただし、条件を満足する仮想プロセッサが存在しない場合には、上記選択したプロセッサのプロセッサ状態表のフィールドに非アクティブ状態を指示する。また、上記選択したプロセッサにおいて、上記選択した仮想プロセッサを走行させることをプロセッサ状態表に指示しておく。次に、ディスパッチ命令群は、上記選択したプロセッサに対して、外部割込みを発生させるために、従来よりあ

るマルチプロセッサ用命令 S I G P (Signal Processor)を発行する。これにより、S I G P 命令が発行されたプロセッサに外部割込みが報告され、その割込み処理プログラムにおいて、次の処理が行われる。すなわち、割込み処理プログラムは、プロセッサ状態表により、そのプロセッサによって走行させる仮想プロセッサを識別し、この仮想プロセッサのタイムスライス値をそのプロセッサのタイマに設定し、その仮想プロセッサの状態追跡領域の内容をプロセッサのレジスタに設定することにより、上記選択した仮想プロセッサを走行状態にする。

ディスパッチ命令群は、上述の処理を繰り返すことにより、プロセッサ状態表に非アクティブ状態が指示されていたプロセッサ上で仮想プロセッサを走行させることができる。

次に、ディスパッチ命令群は、さらにウエイト状態でなくアクティブ状態でもない仮想プロセッサ群の中で、最も走行優先度の高い仮想プロセッサを選択し、その仮想プロセッサのアクティブフ

- 15 -

ラグにアクティブ状態を指示する。ただし、条件を満足する仮想プロセッサが存在しない場合には、この処理を行っているプロセッサのプロセッサ状態表のフィールドに、非アクティブ状態を指示する。条件を満足する仮想プロセッサが存在する場合には、この仮想プロセッサのタイムスライス値をタイマに設定し、その仮想プロセッサの状態追跡領域の内容をプロセッサのレジスタに設定することにより、このプロセッサでの上記選択した仮想プロセッサを走行状態にする。

(2) 第1の仮想プロセッサが走行中に、第2の仮想プロセッサの割込みがハードウェアおよびマイクロプログラムにより直接実行され、この割込みにより、第2の仮想プロセッサのウエイト状態が解除された場合には、割込み処理の延長上で仮想プロセッサ状態追跡機構が起動され、前記(1)の場合と同じような処理により仮想プロセッサが起動される。

(3) 仮想プロセッサが走行開始後、V M C P がこの仮想プロセッサに指定したタイムスライス値

- 16 -

の時間が経過した時の割込み処理をV M C P が介在することなく、ハードウェアおよびマイクロプログラムにより直接実行する場合には、割込み処理の延長上で仮想プロセッサ状態追跡機構が起動され、前記(1)の場合と同じ処理により仮想プロセッサが起動される。

(4) 仮想プロセッサが走行開始後、V M C P がこの仮想プロセッサに指定したタイムスライス値の時間が経過した時の仮想プロセッサ切換処理をV M C P が介在して行う場合、あるいは、前記の(1)(2)で述べたプロセッサをウエイト状態にする命令、または割込み処理をV M C P が介在して行う場合には、V M C P または仮想プロセッサ状態追跡機構、および仮想プロセッサ状態追跡マイクロプログラムは、走行していた仮想プロセッサの状態を該当する状態追跡領域に格納し、該当する仮想プロセッサのアクティブフラグに非アクティブ状態を設定する。次に、V M C P は、制御がV M C P に渡された原因に応じて、タイムスライス値を使い切った場合の処理、プロセッサをウエ

イト状態にする命令のシミュレーション、割込み処理を行った後、本発明による仮想プロセッサ選択・起動用命令を発行する。この命令により、前記(1)で述べたディスパッチ命令群に制御が渡され、選択したプロセッサ上で走行優先度に従つて選択した仮想プロセッサが起動される。

(5) VM走行中に、VM上のOSが、本発明により設定される1組以上の特定命令の命令コードおよび条件コードの組を保持する手段に指定されている命令コードの命令を発行し、しかもその条件コードが上記保持手段の条件コードに一致する場合には、次の処理を行う。一般にマルチプロセッサを構成する計算機下のOSでは、複数のプロセッサが同一のメモリを参照・更新する場合の競合を防ぐために、ロックの手法を用いている。すなわち、一方のプロセッサが特定の命令により特定のメモリにロックをかけると、他のプロセッサはロックをかけたプロセッサがロックを解除するまで、そのメモリの参照・更新はできない。仮想計算機システムでは、実プロセッサが例えば2台

- 19 -

スパシチ命令群は、走行していた仮想プロセッサと同一VM上の仮想プロセッサの状態の判定と選択を行い、前記(1)と同じ方法により、上記仮想プロセッサを走行状態にする。

以上が、本発明による仮想プロセッサの切換え制御方式の処理概要である。各仮想プロセッサの走行優先度は、VMCPがコマンド等により固定的に与えることも可能であり、また仮想プロセッサ走行中のプロセッサ利用率等により動的にVMCPが変更することも可能である。また、走行優先度とタイムスライス値との対応を持つことにより、仮想プロセッサのタイムスライス値を動的に制御することも可能である。

このように、本発明では、少量のハードウェアおよびマイクロプログラムにより、VMCPが指定した走行優先度にもとづき、次に走行する仮想プロセッサの選択と起動を行うことによって、オーバヘッドを削減している。

【発明の実施例】

以下、本発明の実施例を、図面により詳細に説

存在しても、同一VM上の仮想プロセッサが走行しているとは限らない。このため、VMの領域にロックがかかっているために、あるプロセッサ上でOSがロックの解除を待つループする場合には、ロックをかけたVMの仮想プロセッサを走行させ、ロックの解除を行わせることが望ましい。

ロックがかかっているか否かは、特定の命令において、特定の条件コードとなるか否かで判定できる。このため、本発明では、上記命令コードと条件コードとを保持するレジスタを設ける。そして、VM上のOSが上記命令を発行し、その条件コードが上記特定の条件コードの場合には、仮想プロセッサ状態追跡機構が起動され、仮想プロセッサ状態追跡マイクロプログラムとともに、走行中の仮想プロセッサの状態を該当する状態追跡領域に格納し、またその仮想プロセッサのアクティブフラグに非アクティブ状態を設定する。次に、仮想プロセッサ状態追跡機構および仮想プロセッサ状態追跡マイクロプログラムは、主記憶装置上のディスパッチ命令群に制御を渡す。ディ

- 20 -

明する。第1回は、本発明の一実施例を示すVMのブロック図である。第1回において、1は主記憶装置、4は命令実行回路、5は割込み回路、6はプロセッサを制御するプロセッサレジスタ、7は仮想プロセッサの切換え信号を線107を介して割込み回路5に送出するためのタイマ、10は仮想プロセッサ状態追跡機構、35はアドレスレジスタである。

主記憶装置1には、VMS下におけるVMの各仮想プロセッサごとに、仮想プロセッサを走行状態にする時に必要な情報を持つ状態表示テーブル2-1-0~2-n-0が設けられる。仮想プロセッサ管理用テーブル3は、仮想プロセッサの状態表示テーブル2-1~2-nを、走行優先度別にキューイングするためのテーブルである。走行可能仮想プロセッサ管理用テーブル3-1では、iの値が小さいほど、走行優先度の高い状態表示テーブル2-1~2-nがキューイングされる。第1回では、2番目に走行優先度の高い仮想プロセッサ管理用テーブル3-1に3つの状態表示テ

一ブル 2-1-0, 2-1-1 および 2-1-2 がキューイングされており、また最も走行優先度の低い仮想プロセッサ管理用テーブル 3-n には、1つの状態表示テーブル 2-n-0 がキューイングされている。また、プロセッサ状態テーブル 2-8 は、計算機を構成するプロセッサの状態の管理テーブルである。さらに、ディスパッチ命令群 2-9 は、仮想プロセッサおよびプロセッサを選択し、このプロセッサ上で、該当仮想プロセッサの起動を行う命令群である。命令実行回路 4 における 1-2 は命令実行マイクロプログラムであり、割込み回路 5 における 1-3 は割込みマイクロプログラムである。また、プロセッサレジスタ 6 には PSW 8 が設置され、この PSW 8 にはプロセッサがウエイト状態か否かを示すウエイトビット 9 が設けられる。ウエイトビット 9 は、その値が“1”的時、プロセッサがウエイト状態であることを示し、“0”的時、プロセッサがウエイト状態でないことを示す。6-1-6-1 は、それぞれプロセッサレジスタである。仮想プロセッサ状態追跡機構 1

- 23 -

プロセッサがアクティブである、つまりあるプロセッサで走行中か否かを示すアクティブフラグである。2-0 は、該当仮想プロセッサの走行優先度 1-4 を、VMCP が動的に変更するか否かを指示するチエインジビットである。また、3-0 はプロセッサ番号であり、アクティブフラグ 1-9 が“1”的時に、走行中の該当プロセッサ番号を格納する。

第 3 図は、第 1 図におけるプロセッサ状態テーブルの説明図である。プロセッサテーブル 2-8 は、計算機を構成するプロセッサごとに、現在プロセッサがアクティブ状態か否か、またアクティブ状態の場合には、どの仮想プロセッサをアクティブとしているかを表わす。すなわち、プロセッサアクティブビット 3-2-8 が“0”的時には、プロセッサ番号 4 のプロセッサ、つまりプロセッサ番号レジスタ 3-1 に 8 が格納されているプロセッサは、非アクティブ状態であることを表わす。また、プロセッサアクティブビット 3-2-8 が“1”的時には、プロセッサ番号 8 のプロセッサはアクティブ状態であり、またアクティブ状態表示テーブルア

0 における 1-1 は、仮想プロセッサ状態追跡マイクロプログラムである。アドレスレジスタ 3-5 には、走行中の仮想プロセッサ状態表示テーブル 2-1-2-n のアドレスが格納される。また、プロセッサ番号レジスタ 3-1 は、該当プロセッサのプロセッサ番号を保持する。第 1 図では、この場合、状態表示テーブル 2-1-0 に対応する仮想プロセッサが走行中であることを表わす。

第 2 図は、第 1 図における状態表示テーブルの構成を示す図である。第 2 図において、1-4 は該当仮想プロセッサのプロセッサの状態を追跡するための状態追跡領域、1-5 は該当仮想プロセッサの走行優先度、1-6 は該当仮想プロセッサのタイムスライス値である。1-7 は同一走行優先度の状態表示テーブル 2-1-2-n を、仮想プロセッサ管理用テーブル 3 にキューイングするための同一走行優先度状態表示テーブルアドレスポインタ、1-8 は同一 VM 上の仮想プロセッサの状態表示テーブルのアドレスを有する同一 VM 状態表示テーブルアドレスポインタである。1-9 は、その仮想

- 24 -

ドレス 3-4-8 にアドレスが格納されている状態表示テーブル 2 に対応する仮想プロセッサが使用中であることを表わす。また、プロセッサロジックワード 3-3-8 は、プロセッサ状態テーブル 2-8 の該当プロセッサフィールドの参照、および更新を行う場合のロックに用いるフィールドである。

先ず、第 1 の実施例として、走行中の仮想プロセッサ上の OS が、プロセッサをウエイト状態にする命令を発行し、この命令が VMCP の介在なしに、ハードウェアおよびマイクロプログラムにより直接実行された場合の処理を述べる。

走行中の仮想プロセッサ上の OS が PSW 8 をウエイト状態にする、つまりウエイトビット 9 を“1”にする命令を線 1-0-0 を通して命令実行回路 4 に出すと、この命令が命令実行回路 4 および命令実行マイクロプログラム 1-2 により直接実行可能な場合には、線 1-0-1 により PSW 8 のウエイトビット 9 が“1”となる。ウエイトビット 9 が“1”となると、線 1-0-2 により仮想プロセッサ状態追跡機構 1-0 が起動され、仮想プロセッサ状態

追遡マイクロプログラム11とともに、次のようにして仮想プロセッサの状態追遡を行う。

先ず、仮想プロセッサ状態追遡機構10および仮想プロセッサ状態追遡マイクロプログラム11は、線121からアドレスレジスタ35の値を取り込み、この値をアドレスとする状態表示テーブル2-1-0の状態追遡領域14に、プロセッサレジスタ6の内容を線102、線103により追遡する。また、線122、線103によりタイム7の値を、状態表示テーブル2-1-0のタイムスライス値16に追遡する。また、線103により状態表示テーブル2-1-0のアクティブフラグ19に、“0”を設定する。

次に、仮想プロセッサ状態追遡機構10および仮想プロセッサ状態追遡マイクロプログラム11は、主記憶装置1上のディスパッチ命令群29に制御を渡す。これは、例えば、ディスパッチ命令群29のアドレスを、特定のプロセッサレジスタ6のレジスタに設けておくことにより可能である。また、プロセッサ状態テーブル28および仮想ブ

ロセッサ管理用テーブル3についても、アドレスが特定のプロセッサレジスタ6のレジスタに設定されている。ディスパッチ命令群29の処理は、次のようにして行われる。先ず、プロセッサ状態テーブル28のプロセッサロジックワード33-mに対してロックをかけた後、プロセッサアクティブビット32-mが“1”か否かを判定する。プロセッサがアクティブ、つまりプロセッサアクティブビット32-mが“1”であれば、ロックを解除して、次のプロセッサのフィールドを判定する。プロセッサが非アクティブ状態、つまりプロセッサアクティブビット32-mが“0”であれば、プロセッサアクティブビット32-mに“1”を設定後、ロックを解除する。以上の処理により、アクティブでないプロセッサを選択する。次に、仮想プロセッサ管理用テーブル3にキューリングされたウエイト状態でなく、かつアクティブ状態でない状態表示テーブル2のうちで、最も走行優先度が高いものを選択し、状態表示テーブル2のアクティブフラグ19に“1”を設定する。状態表示テ

- 27 -

ーブル2に対応する仮想プロセッサがウエイト状態か否かは、状態追遡領域14中にあるPSW8のウエイトビット9に対応するビットにより判定できる。すなわち、このビットが“1”であれば、該当する仮想プロセッサはウエイト状態であり、“0”であればウエイト状態でない。ただし、条件を満足する仮想プロセッサが存在しない場合には、プロセッサ状態テーブル28の上記選択したプロセッサのプロセッサアクティブビット32を“0”にする。

いま、状態表示テーブル2-1-1に対応する仮想プロセッサがウエイト状態でなく、アクティブフラグ19が“0”であるとする。これにより、ディスパッチ命令群29は、プロセッサ状態テーブル28の上記選択したプロセッサのアクティブ状態表示テーブルアドレス34に、上記選択した仮想プロセッサの状態表示テーブル2-1-1のアドレスを設定する。また、状態表示テーブル2-1-1のプロセッサ番号30に、上記選択したプロセッサのプロセッサ番号を設定する。ここで、

- 28 -

プロセッサ番号は、プロセッサ状態テーブル28のインデックスに等しいものとする。すなわち、プロセッサ状態テーブル28-mに対応するプロセッサのプロセッサ番号は、mとする。次に、ディスパッチ命令群29は、従来よりマルチプロセッサ支援用にあるSIGP命令を上記選択したプロセッサに対して発行する。SIGP命令は、オペランドとしてプロセッサ番号と、副指令コードを持つが、プロセッサ番号には、上記選択したプロセッサのプロセッサ番号を設定し、副指令コードには、仮想プロセッサの起動の指示を行う。SIGP命令により、上記選択したプロセッサに対して、外部割込みが発生するとともに、副指令コードが送られる。この外部割込みにより、次の処理が行われ、そのプロセッサが起動される。

外部割込みの処理では、先ずプロセッサ番号レジスタ読み取り命令を発行する。この命令により、命令実行回路4と命令実行マイクロプログラム12は、線100、線104によりプロセッサ番号レジスタ31の値をプロセッサレジスタ6内のレ

ジスタ、または上記憶上の指定された領域に格納する。次に、プロセッサ状態テーブル2-8の上記命令により得られたプロセッサ番号に対応するプロセッサフィールドのアクティブ状態表示テーブルアドレス3-4にアドレスが格納される状態表示テーブル2-1-1に対応する仮想プロセッサを、そのプロセッサにおいて走行させる処理を行う。すなわち、線100、線120により、アドレスレジスタ3-5に状態表示テーブル2-1-1のアドレスを設定し、タイムスライス値1-6を線100、線108によりタイマ7へ。また線100、線101により状態追跡領域1-4の内容をプロセッサレジスタ6に設定する。以上の処理により、状態表示テーブル2-1-1に対応する仮想プロセッサの走行を開始する。

以上述べた処理を繰り返すことにより、該当プロセッサ以外の非アクティブ状態のプロセッサにおいて、走行の条件を満足する仮想プロセッサを走行させることができる。また、ディスパッチ命令群2-9は、前述の方法により、ウェイト状態で

なく、かつアクティブ状態でない仮想プロセッサ群の中で、最も走行優先度の高い仮想プロセッサを選択し、その状態表示テーブル2のアクティブフラグ1-9に“1”を設定する。ただし、条件を満足する仮想プロセッサが存在しない場合には、ディスパッチ命令群2-9は、プロセッサ番号レジスタ読み取り命令により、前述のように、そのプロセッサのプロセッサ番号をプロセッサ番号レジスタ3-1から、線100、線104により読み出し、プロセッサ状態表2-8の該当するフィールドのプロセッサアクティブビット3-2に、非アクティブ状態を示す“0”を設定する。条件を満足する仮想プロセッサが存在する場合には、選択した状態表示テーブル2のプロセッサ番号3-0に上記プロセッサ番号を格納する。また、上記選択した状態表示テーブル2のタイムスライス値1-6を、線100、線108により命令実行回路4および命令実行マイクロプログラム1-2によってタイマ7へ。また、状態追跡領域1-4の内容を線100、線101により命令実行回路4および命令実行マイク

- 31 -

- 32 -

ロプログラム1-2によってプロセッサレジスタ6に設定することにより、このプロセッサで、上記選択した仮想プロセッサを走行状態にする。

次に、第2の実施例として、第1の仮想プロセッサが走行中に、VMCPを介在させることなく削込み回路5および削込みマイクロプログラム1-3により直接処理可能な削込みが発生し、この削込みにより第2の仮想プロセッサのウェイト状態が解除された場合の処理について述べる。

削込みが、削込み回路5および削込みマイクロプログラム1-3により第2の仮想プロセッサに線105によって反映され、その結果、第2の仮想プロセッサのウェイト状態が解除された場合、すなわち、第2の仮想プロセッサの状態表示テーブル2内にある状態追跡領域1-4のPSWを表わすフィールドのウェイトビット9に相当する部分が“1”から“0”に変わった場合、線106により仮想プロセッサ状態追跡機構1-0に起動信号を送出する。その後、第1の実施例で述べた場合と全く同じ処理により、仮想プロセッサを走行させる。

次に、第3の実施例として、仮想プロセッサが走行開始後、VMCPが介在することなく、ハードウェアおよびマイクロプログラムにより直接実行する場合の処理について述べる。

この場合にも、第2の実施例と同じように、削込み回路5および削込みマイクロプログラム1-3は、線105により仮想プロセッサ状態追跡機構1-0に起動信号を送る。その後、第1の実施例で述べた場合とほぼ同じ方法により、仮想プロセッサのディスパッチ処理を行うが、次の2つの点が異なる。

第1番目は、仮想プロセッサ状態追跡機構1-0および仮想プロセッサ状態追跡マイクロプログラム1-1が、該当する状態表示テーブル2のタイムスライス値1-6に設定する値は、タイマ7の値ではなく、例えば特定のプロセッサレジスタ6に設定されている正の値を線102、線103により設定する。

第2番目は、走行していた仮想プロセッサの状態表示テーブル2を現在キューイングされている

仮想プロセッサ管理用テーブル3-1の1番最後にキューイングする。以上の変更により、仮想プロセッサを走行させる。

次に、本発明の第4の実施例について、仮想プロセッサの選択・起動用命令に関する動作を説明する。VMCPが仮想プロセッサの切換処理を行う場合としては、仮想プロセッサが走行開始後、その仮想プロセッサに割当てられたタイムスライス値に示された時間が経過した時の割込み処理をVMCPが介在して行う場合、OSが発行したPSWをエイト状態にする命令のシミュレーションを、VMCPが行う場合、およびVMCPによる割込みのシミュレーションにより、仮想プロセッサのエイト状態が解除された場合等が考えられる。

本発明による仮想プロセッサ選択・起動用命令は、上記の場合の仮想プロセッサの切換処理を高適化する。

仮想プロセッサの走行が中断されると、VMCPまたは仮想プロセッサ状態追跡機構10および

仮想プロセッサ状態追跡マイクロプログラム11により、走行していた仮想プロセッサの状態を線102、線103により該当する状態表示テーブル2の状態追跡領域14に格納し、次に線122、線103によりタイマ7の値を該当するタイムスライス値に追加する。また、該当仮想プロセッサのアクティブフラグ19に、非アクティブ状態を表わす“0”を設定する。次に、VMCPは、制御がVMCPに渡された原因に応じて、タイムスライス値を使い切った場合の処理、プロセッサをウエイト状態にする命令のシミュレーション、割込み処理を行う。その後、VMCPは、仮想プロセッサ選択・起動用命令を発行する。VMCPが仮想プロセッサ選択・起動用命令を発行すると、線100を通して命令実行回路4に仮想プロセッサ選択・起動用命令の起動信号が出される。仮想プロセッサ選択・起動用命令が発行されると、命令実行回路4および命令実行マイクロプログラム12は、線101によりディスパッチ命令群29のアドレスを有するプロセッサレジスタ6-1の値

- 35 -

を読み出し、主記憶装置1上のディスパッチ命令群29に制御を渡す。ディスパッチ命令群29の処理は、第1の実施例と全く同じである。これにより、走行可能な仮想プロセッサを走行状態にすることができる。

次に、第5の実施例として、仮想プロセッサ1が走行中に同一VMの仮想プロセッサ2がロックをかけている領域を参照・更新しようとする場合の処理を述べる。

第4図は、本発明の他の実施例を示すVMSのプロック図である。第4図において、第1図と異なる点は、命令コード等のレジスタ21～24および演算器25、26を命令実行回路4に接続したことである。21は、実行中の命令の命令コードを保持する命令コードレジスタ、22は命令の条件コードを保持する条件コードレジスタ、23は特定の命令の命令コードを有する命令コード保持レジスタ、24は特定の条件コードを有する条件コード保持レジスタ、25、26は比較回路、27はアンド回路である。

- 36 -

OSが命令を発行すると、線100により命令実行回路4に命令起動信号が出る。命令実行回路4および命令実行マイクロプログラム12は、条件コードレジスタ22を線115によりイニシャライズした後、命令コードレジスタ21に線113により該当する命令コードをセットする。また、上記命令の条件コードが決定した時に、命令実行回路4は、条件コードレジスタ22に、この条件コードを設定するものとする。本実施例では、命令コードレジスタ21と条件コードレジスタ22を1個ずつ設けているが、複数個持たせることも勿論可能である。

第5図は、本発明における状態表示テーブルのキューイング方法の説明図である。第5図において、状態表示テーブル2-1-0と2-n-0は、同一VMの異なる仮想プロセッサに対応するものであり、互いに同一VM状態表示テーブルアドレスポインタ18(第2図参照)によりポイントされている。これは、同一VMの全ての仮想プロセッサに対応する状態表示テーブル2を、仮想プロ

セシサがウエイト状態であるか否かにかかわらず、リンクするものとする。

命令コード保持レジスタ23と条件コード保持レジスタ24には、それぞれOSがある領域にロックがかかっているか否かを判定する時に使用する命令の命令コードと、上記命令において、他プロセシサのロックがかかっていた時にPSW8に設定される条件コードをVMCP専用命令により線100, 112, 114により、予め設定しておく。

状態表示テーブル2-1-0に対応する走行中の仮想プロセシサ1のOSが、命令コード保持レジスタ23に命令コードを有する命令を発行し、その条件コードが条件コードレジスタ24に示された条件コードの場合、線116, 117, 118, 119から信号が比較回路25および26に送られ、次に線109、110から信号がアンドゲート27に送られ、線111により仮想プロセシサ切換制御機構(状態遮断機構)10が起動される。仮想プロセシサ切換制御機構10および仮

想プロセシサ切換制御マイクロプログラム(状態遮断マイクロプログラム)11は、プロセシサレジスタ6の値を、線102から取り込み、これをアドレスレジスタ35にアドレスが格納されている状態表示テーブル2-1-0の状態遮断領域14に、線103により遮断する。次に、線122、線103により、タイマ7の値をタイムスライス値16に格納する。また、状態表示テーブル2-1-0のアクティブフラグを"0"に設定する。

次に、仮想プロセシサ切換制御機構10および仮想プロセシサ切換制御マイクロプログラム11は、主記憶装置1上のディスパッチ命令群29に制御を渡す。ディスパッチ命令群29の処理は、第1の実施例で述べた処理とほぼ同じであるが、走行させる対象とする仮想プロセシサの選択方法が異なる。本実施例では、ディスパッチ命令群29は、直前に走行していた仮想プロセシサを除く、仮想プロセシサと同一VM上の仮想プロセシサでウエイト状態でなく、かつアクティブ状態でない仮想プロセシサを、同一VM状態表示テーブルア

- 39 -

- 40 -

ドレスポインタ18を用いて選択する。これにより、中断された仮想プロセシサでない同一VM上の仮想プロセシサを走行させることができる。

なお、本実施例では、走行していた仮想プロセシサを一旦非アクティブ状態としたが、アクティブ状態を継続し、他プロセシサにおいて仮想プロセシサのディスパッチ処理終了後、上記走行していた仮想プロセシサを継続して走行させることも可能である。また、以上の実施例においては、ディスパッチ命令群29をソフトウエアにより実現したが、マイクロプログラムでも実現可能である。

最後に、状態表示テーブル2内の走行優先度15の設定方法と、VMまたはVMCPがマルチプロセシサ構成時における仮想プロセシサ切換処理について、述べる。

走行優先度14は、VMCPがコマンド等により固定的に与えることも可能であり、またVMCPが動的に変更することも可能である。例えば、チエインジビット20が"1"の時には、コマンドによってのみ走行優先度15を変更可能とし、チ

エインジビット20が"0"の時には、VMCPは該当仮想プロセシサが走行時のCPU利用率等をもとにし、走行優先度15を動的に変えることは、従来の技術により可能である。

また、走行優先度15に対応してタイムスライス値16を変更する場合には、走行優先度15とタイムスライス値16との対応表を設け、走行優先度15が変わる度に、その対応表により、タイムスライス値16を更新すればよい。

また、本実施例においては、説明を省略したがVMあるいはVMCPがマルチプロセシサ構成の場合には、主記憶1上の仮想プロセシサ管理用テーブル3、状態表示テーブル2、およびプロセシサ状態テーブル28を参照・更新する場合には、通常のマルチプロセシサ技術におけるロックにより、予め他のプロセシサあるいは仮想プロセシサからは、参照・更新できないようにしておく。

【発明の効果】

以上説明したように、本発明によれば、VM上の仮想プロセシサがウエイト状態になつたり、同

—VM上他の仮想プロセッサのロックによりOSがループ状態となる場合には、その仮想プロセッサが走行しているプロセッサ、およびアクティブ状態にないプロセッサ上で、走行可能な仮想プロセッサを起動させることができるので、マルチプロセッサの仮想計算機において、計算機のプロセッサを効率よく利用することができる。

4. 図面の簡単な説明

第1図は本発明の一実施例を示すVMSのブロック図、第2図は第1図における状態表示テーブルの説明図、第3図は第1図におけるプロセッサ状態テーブルの説明図、第4図は本発明の他の実施例を示すVMSのブロック図、第5図は本発明における状態表示テーブルのキューリング方法の説明図である。

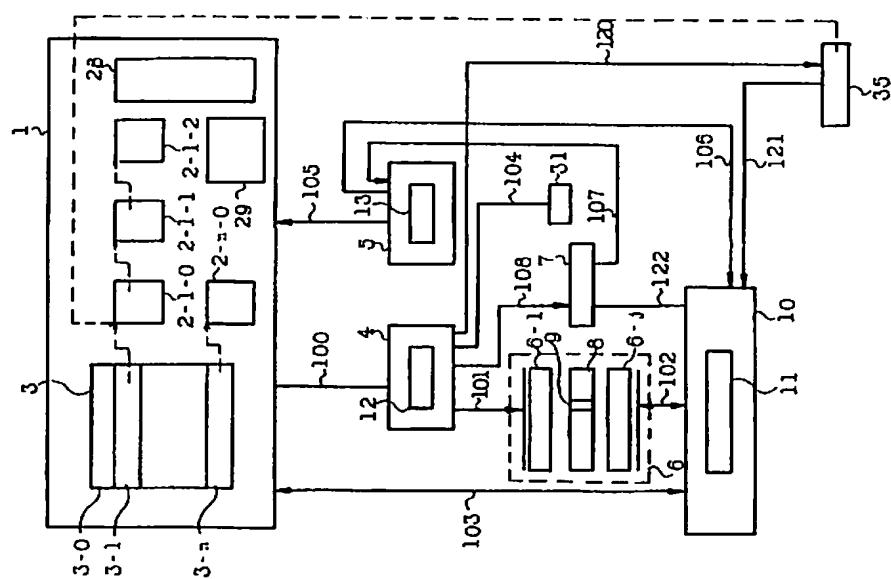
1: 主記憶装置、2: 状態表示テーブル、3: 仮想プロセッサ管理用テーブル、4: 命令実行回路、5: 割込み回路、6: プロセッサレジスタ、7: タイマ、8: PSW、9: ウエイトビット、10: 仮想プロセッサ状態追還機構、11: 仮想

プロセッサ状態追還マイクロプログラム、12: 命令実行マイクロプログラム、13: 割込みマイクロプログラム、14: 状態追還領域、15: 走行優先度、16: タイムスライス値、17: 同一走行優先度状態表示テーブルアドレスポインタ、18: 同一VM状態表示テーブルアドレスポインタ、19: アクティブフラグ、20: チエインジビット、21: 命令コードレジスタ、22: 條件コードレジスタ、23: 命令コード保持レジスタ、24: 條件コード保持レジスタ、25, 26: 比較回路、27: アンドゲート、28: プロセッサ状態テーブル、29: デイスバッチ命令群、30: プロセッサ番号、31: プロセッサ番号レジスタ、32: プロセッサアクティブビット、33: プロセッサロジックワード、34: アクティブ状態表示テーブルアドレス、35: アドレスレジスタ、100~122: 信号線。

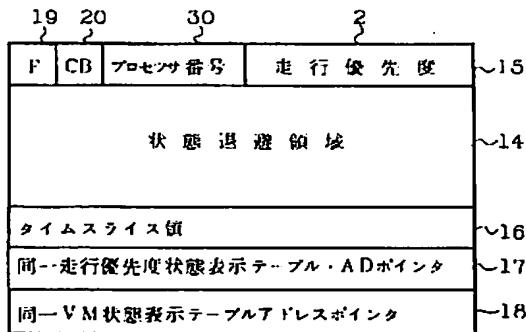
特許出願人 株式会社 日立製作所
代理人弁理士 碓村 駿



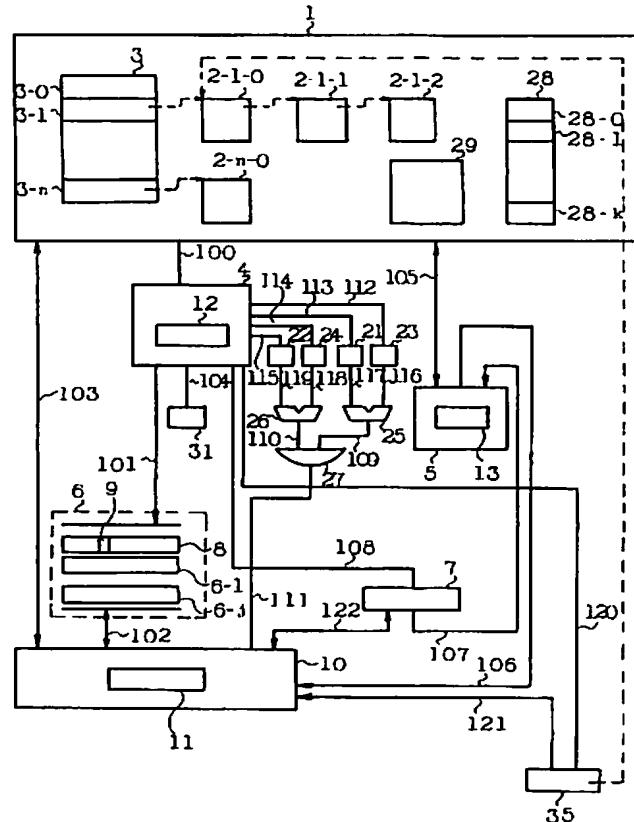
図
1
構
成



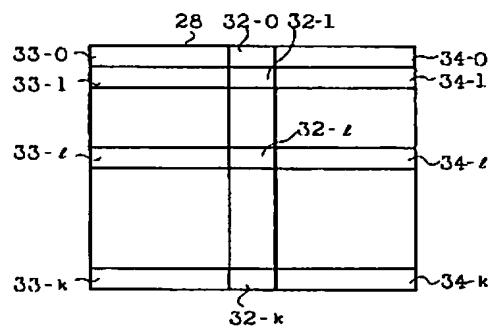
第 2 図



第 4 図



第 3 図



第 5 図

